



19 BUNDESREPUBLIK  
DEUTSCHLAND



DEUTSCHES  
PATENT- UND  
MARKENAMT

12 Offenlegungsschrift  
10 DE 100 65 419 A 1

51 Int. Cl. 7:  
G 05 B 19/04  
G 05 B 19/05

21 Aktenzeichen: 100 65 419.3  
22 Anmeldetag: 27. 12. 2000  
43 Offenlegungstag: 18. 7. 2002

DE 100 65 419 A 1

71 Anmelder:  
Siemens AG, 80333 München, DE

72 Erfinder:  
Amrhein, Armin, 92245 Kümmersbruck, DE; Birzer,  
Johannes, 92551 Stulln, DE; Kiesel, Martin, 91099  
Poxdorf, DE; Schmitt, Regina, 91074  
Herzogenaurach, DE

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

Prüfungsantrag gem. § 44 PatG ist gestellt

54 Industrielle Steuerung mit taktsynchronem Ablaufebenenmodell

57 Für jeweils unterschiedliche Steuerungsaufgaben und unterschiedliche Randbedingungen bzw. Anforderungen des zugrunde liegenden technischen Prozesses wird in einfacher und flexibler Weise ein Ablaufebenenmodell für eine industrielle Steuerung (S) vorgeschlagen, das sowohl SPS- als auch MC-Funktionalität zur Verfügung stellt und somit auch für die Steuerung von Produktionsmaschinen geeignet ist.

Synchron getaktete Ebenen
Anwenderebene für Systemexceptions
ereignisgesteuerte Anwenderebene
zeitgesteuerte Anwenderebene
sequentielle Anwenderebene
zyklische Anwenderebene

Priorität

DE 100 65 419 A 1

[0001] Die Erfindung bezieht sich auf eine industrielle, mit einem Runtime-System ausgestattete Steuerung, insbesondere für Produktionsmaschinen, wobei das Runtime-System ein Ablaufebenenmodell enthält, das mehrere Ablaufebenen unterschiedlichen Typs mit unterschiedlicher Priorität aufweist, wobei eine Ebenengruppe des Runtime-Systems aus einem Grundtakt abgeleitete synchron getaktete Ebenen enthält.

[0002] Ferner bezieht sich die Erfindung auf die Erstellung eines Runtime-Systems einer industriellen Steuerung mit synchron getakteten System- und Anwenderebenen.

[0003] Es ist heutzutage üblich, sowohl für die speicherprogrammierbare Steuerung (SPS) als auch für die Bewegungssteuerung (MC) jeweils unterschiedliche hierarchische Ablaufebenen zu modellieren, denen Software-Tasks zur Steuerung des jeweiligen technischen Prozesses zugeordnet werden. Diese Tasks können Systemaufgaben erfüllen, sie können aber auch anwenderprogrammiert sein.

[0004] Aus DE 197 40 550 A1 ist es bekannt, dass Prozesssteuerungsfunktionalitäten der speicherprogrammierbaren Steuerungen "SPS" und Bewegungsfunktionalitäten von MC-Steuerungen in einem einheitlichen konfigurierbaren Steuerungssystem integriert werden können.

[0005] Diese SPS/MC-Integration geschieht in Form des Zusammenschaltens von SPS- und MC-Steuerungsbaugruppen. Bei einer solchen Ausführung der Integration wird aber keine optimale und effiziente Taskstruktur für die Gesamtheit der Steuerungsaufgaben erreicht. Außerdem werden bei dieser Art der Integration hauptsächlich die klassischen MC-Funktionalitäten, wie sie insbesondere für Werkzeugmaschinen relevant sind, unterstützt. Anforderungen an die Steuerung, wie sie aus dem Betrieb von Produktionsmaschinen bekannt sind, werden durch diese Art des Zusammenschaltens von SPS- und MC-Steuerungsbaugruppen nicht optimal unterstützt.

[0006] In der Anmeldung DE 199 31 933.2 wird vorgeschlagen, den Takt des Kommunikationssystems zwischen dem PC-System und den peripheren Geräten für einen Wechsel zwischen einem Echtzeitbetriebsprogramm und einem Nicht-Echtzeitbetriebsprogramm heranzunehmen. Hier ist es aber die Aufgabe dieser Taktabgreifung aus dem Kommunikationssystem, in einem industriellen Prozess einen möglichst reibungslosen Wechsel zwischen Echtzeit- und Nicht-Echtzeitanwendungen stattfinden zu lassen. Bei dieser Ausgestaltung wird der Grundtakt aber nur aus dem Takt des Kommunikationsmediums abgeleitet und er wird nur für den Wechsel des Betriebssystemmodus eines PC-Systems verwendet.

[0007] Der Erfindung liegt daher die Aufgabe zugrunde, für jeweils unterschiedliche Steuerungsaufgaben und unterschiedliche Randbedingungen bzw. Anforderungen des zugrunde liegenden technischen Prozesses in einfacher Weise optimale Ausprägungen einer industriellen Steuerung zu erstellen, die sowohl SPS- als auch MC-Funktionalität zur Verfügung stellt und somit auch für die Steuerung von Produktionsmaschinen geeignet ist.

[0008] Diese optimalen Ausprägungen werden prinzipiell durch ein einheitliches konfigurierbares Ablaufebenenmodell für die Steuerungs-Tasks der industriellen Steuerung erreicht.

[0009] Von diesem Ansatz ausgehend wird die oben genannte Aufgabe dadurch gelöst, dass das Runtime-System der industriellen Steuerung ein Ablaufebenenmodell enthält, das mehrere Ablaufebenen unterschiedlichen Typs mit unterschiedlicher Priorität aufweist, wobei folgende Ablaufebenen vorgesehen sind:

- a) eine Ebenengruppe mit aus einem Grundtakt abgeleiteten synchron getakteten Ebenen, bestehend aus mindestens einer Systemebene und mindestens einer Anwenderebene, wobei die Ebenen dieser Ebenengruppe untereinander eine Priorisierung aufweisen können,
- b) einer Anwenderebene für Systemexceptions,
- c) einer ereignisgesteuerten Anwenderebene,
- d) einer zeitgesteuerten Anwenderebene,
- e) einer sequentiellen Anwenderebene,
- f) einer zyklischen Anwenderebene,

wobei Anwenderebenen der Ebenengruppe a) optional synchron zu einer der Systemebenen der Ebenengruppe a) laufen können.

[0010] Ein wesentlicher Vorteil dieser Schichtung liegt darin, dass die Kommunikation zwischen den Tasks der Prozesssteuerung (SPS) und denen der Bewegungssteuerung (MC) minimiert wird. Ein weiterer Vorteil liegt darin, dass die Programmierung der Steuerungsaufgaben für die Prozesssteuerung und für die Bewegungssteuerung in einer einheitlichen Programmiersprache mit einer einheitlichen Erstelloberfläche erfolgen kann und dass der Anwender ein für seine jeweiligen Anforderungen zugeschnittenes Ablaufebenenmodell flexibel erstellen kann.

[0011] Eine erste Ausgestaltung der vorliegenden Erfindung liegt darin, dass der Grundtakt des Ablaufebenenmodells aus einem internen Timer oder aus einem internen Takt eines Kommunikationsmediums oder aus einem externen Gerät oder von einer Größe, die zum technologischen Prozess gehört, ableitbar ist. Dadurch kann sehr flexibel und sehr einfach der Grundtakt für das Ablaufebenenmodell abgeleitet werden. Dadurch, dass der Grundtakt für das Ablaufebenenmodell auch von einer Größe, die zum technologischen Prozess gehört, ableitbar ist, kann auf eine sehr einfache Weise eine direkte Rückkopplung aus dem technologischen Prozess zur Steuerung erhalten werden.

[0012] Eine weitere vorteilhafte Ausgestaltung der Erfindung liegt darin, dass die ereignisgesteuerte Anwenderebene, die zeitgesteuerte Anwenderebene, die sequentielle Anwenderebene, die zyklische Anwenderebene und die Anwenderebene für System Exceptions optional sind. Der Anwender kann sich dadurch sehr flexibel eine Steuerung erstellen, die für seine konkreten vorliegenden Anforderungen sehr effizient ist und die die gerade benötigten Ablaufebenen enthält und somit keinen unnötigen Overhead beinhaltet.

[0013] Eine weitere vorteilhafte Ausgestaltung der Erfindung liegt darin, dass die synchronen Ebenen zum Grundtakt übersetzt und/oder unteretzt und/oder im Verhältnis 1 : 1 getaktet sind. Dadurch können die Ebenen jeweils sehr leicht zu einem Vielfachen des Grundtaktes getaktet werden oder aber auch jeweils zu einem reziproken Vielfachen getaktet werden. Ausgehend von einer gemeinsamen Ausgangsgröße können somit sehr einfach Übersetzungen, aber auch Untersetzungen für die jeweiligen Ebenen erreicht werden.

[0014] Eine weitere vorteilhafte Ausgestaltung der Erfindung liegt darin, dass weitere priorisierende Schichtungen innerhalb der Ablaufebenen vorgesehen sind. Die Softwarestruktur der industriellen Steuerung lässt sich dadurch optimal an die unterschiedlichen Steuerungsaufgaben bzw. an die Anforderungen des zugrunde liegenden technischen Prozesses anpassen. Somit lassen sich z. B. unterschiedliche Fehlerursachen unterschiedlichen Ebenen mit z. B. aufsteigender Priorität zuordnen.

[0015] Eine weitere vorteilhafte Ausgestaltung der Erfindung liegt darin, dass optional Anwender-Tasks beim Systemhochlauf und/oder beim Systemrunterfahren durchlauf-

bar sind. Dadurch können beim Systemhochlauf z. B. Initialisierungsfunktionen angestoßen werden oder beim Systemrunterfahren kann durch Anwenderprogrammierung sichergestellt werden, dass die im System vorhandenen Achsen eine definierte Position einnehmen.

[0016] Eine weitere vorteilhafte Ausgestaltung der Erfindung liegt darin, dass in die Anwendererebenen Anwenderprogramme ladbar sind. Dadurch kann der Anwender in seinen Anwenderprogrammen die Funktionalität der Steuerung sehr flexibel an die zugrunde liegenden Anforderungen des technischen Prozesses anpassen und er kann auch die Anwenderprogramme in unterschiedliche Anwendererebenen laden, um dadurch eine für seine jeweiligen Applikationen effektive Ausprägung der Steuerung zu erreichen.

[0017] Eine weitere vorteilhafte Ausgestaltung der Erfindung liegt darin, dass in die Anwendererebenen Anwenderprogramme ladbar sind, die je nach Typ der Anwendererebene zyklusorientiert oder sequentiell programmiert sind. Der Anwender kann somit in ein einheitliches Ablaufebenenmodell bzw. Runtime-System einer industriellen Steuerung sowohl zyklusorientierte Anwenderprogramme (z. B. typische SPS-Funktionalität) als auch sequentielle bzw. ablauforientierte Anwenderprogramme (z. B. typische Bewegungsfunktionalität) laden. Ein Anwender kann somit nach unterschiedlichen Paradigmen programmierte Programme (zyklusorientiert für SPS-Funktionalität und sequentiell bzw. ablauforientiert für Bewegungsfunktionalität) sehr flexibel und konform in die Anwendererebenen des Ablaufebenenmodells hinzuladen. Zyklusorientierte Programme sind typischerweise zykluszeitüberwacht.

[0018] Eine weitere vorteilhafte Ausgestaltung der Erfindung liegt darin, dass durch das Hinzuladen von Technologiepaketen in die Systemebenen die technologische Funktionalität der Steuerung erweiterbar ist. Damit kann der Anwender ausgehend von einem Basissystem der Steuerung den Befehlsvorrat dieses Basissystems dynamisch zugeschnitten auf die jeweiligen Erfordernissen des zugrundeliegenden technologischen Prozesses oder der Steuerungsaufgabe erweitern. Das Basissystem bildet hierbei den Auslieferungsumfang des Run-Time-Systems einer Steuerung, nämlich ein Echtzeitbetriebssystem, ein Ablaufsystem (mit System- und Anwendererebenen), Sprachbefehle, den SPS-Befehlsvorrat sowie Kommunikations- (z. B. LAN, E/A) und technologische Schnittstellen (z. B. Antriebe, Geber) zum technischen Prozeß. Im Basissystem befindet sich somit die notwendige Grundfunktionalität einer Steuerung. Das Basissystem ist dabei auf unterschiedlichsten HW-Plattformen (z. B. PC-based, Drive-based, Controller-based) ablauffähig. Durch das Zuladen von Technologiepaketen hat ein Anwender somit die Möglichkeit einer Skalierung des Run-Time-Systems der Steuerung.

[0019] Eine weitere vorteilhafte Ausgestaltung der Erfindung liegt, dass Mechanismen bereitgestellt sind, die es einem Anwender ermöglichen, im Programmzyklus auf eine beliebige Bedingung zu warten, wobei bei Erfülltsein der Bedingung der Programmfluss unmittelbar fortsetzbar ist, bei Nichterfülltsein der Bedingung der Programmfluss so lange angehalten wird, bis das Erfülltsein der Bedingung festgestellt wird, wobei beim Warten auf das Erfülltsein der Bedingung die Priorität der Bedingungsüberprüfung im Vergleich zur aktuellen Taskpriorität erhöht wird. Durch diesen Mechanismus wird es ermöglicht, eine zusammengehörige und geschlossene Aufgabenstellung in einem Stück Code eines Anwenderprogrammes auszudrücken, ohne dass weitere Mechanismen wie z. B. Event-Handler erforderlich sind. Dadurch wird einerseits in der Steuerung Verwaltungs-Overhead vermieden, was direkt die System-Performance erhöht und andererseits aus programmiertechnischer Sicht

das Lokalisierungsprinzip unterstützt, wodurch z. B. das Debugging erleichtert wird.

[0020] Der beschriebene Mechanismus und der dazugehörige Befehl werden im weiteren als "wait\_for-condition" bezeichnet.

[0021] Eine weitere vorteilhafte Ausgestaltung der Erfindung liegt darin, dass für die Formulierung der Bedingungen Prozesssignale und/oder interne Signale der Steuerung und/oder Variablen aus Anwenderprogrammen verwendbar sind. Dadurch ist es dem Anwender möglich, bei der Beschreibung der Bedingungen Prozesssignale, interne Steuerungssignale oder Anwendervariablen in einer einheitlichen Weise zu verbinden.

[0022] Eine weitere vorteilhafte Ausgestaltung der Erfindung liegt darin, dass verschiedene Ablaufebenen zur Verfügung stehen, die in Vielfachen eines Grundtaktes abgearbeitet werden, wobei technologische Funktionen den Ablaufebenen frei zugeordnet werden. Achsen und technologische Abläufe erfordern unterschiedliche Genauigkeit und Reaktionsfähigkeit. Sie können daher unterschiedlichen Taktsternen (z. B. eine Achse wird im 1 ms Takt gerechnet, eine andere im 2 ms Takt). Im Sinne einer geringeren Gesamtbelastung des Runtime-Systems bzw. des gesamten Systems werden durch die vorliegende Ausgestaltung die entsprechenden Tasks nur so hochprior bearbeitet, wie notwendig.

[0023] Eine weitere vorteilhafte Ausgestaltung der Erfindung liegt darin, dass das Runtime-System der Steuerung ein Ablaufebenenmodell enthält, das mehrere Ablaufebenen unterschiedlichen Typs mit unterschiedlicher Priorität aufweist, wobei folgende Ablaufebenen vorgesehen sind:

- a) Eine Ebene, die bei Technologiefehlern aktiviert wird,
- b) eine Ebene, die bei Zykluszeitüberwachung der Hintergrund-Tasks aktiviert wird,
- c) eine Ebene, die bei Programmfehlern aktiviert wird,
- d) eine Ebene, die bei Systemfehlern aktiviert wird,
- e) eine Ebene, die bei Fehlern oder Alarmen in der Peripherie aktiviert wird,
- f) eine Ebene, die beim Ansprechen von Zeitüberwachungen aktiviert wird,
- g) mindestens eine Ebene, die beim Erkennen von Anwender-Interrupts aktiviert wird,
- h) eine Interpolatorebene,
- i) mindestens eine Ebene für Timer-Interrupts,
- j) mindestens eine Ebene für die, vorzugsweise sequentielle, Abarbeitung von Bewegungs-Tasks und
- k) eine Ebene für die Hintergrund-Bearbeitung,

wobei weitere Anwender-Tasks beim Systemhochlauf und/oder beim Systemrunterfahren durchlaufen werden. Durch diese Ausgestaltung der Ablaufebenenstruktur wird insbesondere ein sehr effizientes Runtime-System für industrielle Steuerungen für Produktionsmaschinen erreicht.

[0024] Eine weitere vorteilhafte Ausgestaltung der Erfindung liegt darin, dass der Grundtakt des Ablaufebenenmodells aus einem internen Timer oder aus einem internen Takt eines Kommunikationsmediums oder von einer Größe, die zum technologischen Prozess gehört, abgeleitet wird. Ein Anwender hat somit große Flexibilität, den Grundtakt zu erzeugen. Der Grundtakt kann z. B. aus äquidistanten Bussystemen wie z. B. Profibus sehr leicht abgeleitet werden.

[0025] Gemäß der Erfindung wird die oben genannte Aufgabe für ein Verfahren der eingangs genannten Art durch die folgenden aufeinander folgenden Schritte gelöst:

- a) Taktgenerierung aus einem internen Hardware-Takt oder einem Kommunikationssystem oder einer Größe

eines technologischen Prozesses,

- b) Bereitstellen von System- und Anwenderebenen für das Runtime-System, die zu diesem Grundtakt synchronisierte Ablaufebenen bereitstellen,
- c) Bereitstellen mindestens einer Anwenderebene, die synchron zu der bzw. den Systemebenen integriert ist und
- d) Programmieren der Anwenderebene bzw. der Anwenderebenen.

[0026] Ein Anwender kann dadurch auf eine definierte Art und Weise ein Runtime-System einer industriellen Steuerung mit synchron getakteten System- und Anwenderebenen erstellen, das Prozesssteuerungsfunktionalitäten von speicherprogrammierbaren Steuerungen "SPS" und Bewegungsfunktionalitäten von MC-Steuerungen integriert, wobei die dazugehörige Taktgenerierung sehr einfach erreicht wird.

[0027] Ein Ausführungsbeispiel der Erfindung ist in der Zeichnung dargestellt und wird im folgenden erläutert.

[0028] Dabei zeigen:

[0029] Fig. 1 die wesentlichen Ablaufebenen einer klassischen speicherprogrammierbaren Steuerung,

[0030] Fig. 2 die wesentlichen Ablaufebenen einer Bewegungssteuerung,

[0031] Fig. 3 die erfindungsgemäße industrielle Steuerung,

[0032] Fig. 4 das Ablaufebenenmodell dieser industriellen Steuerung,

[0033] Fig. 5 ein Ausführungsbeispiel für das Zuladen von Anwenderprogrammen in die Anwenderebenen,

[0034] Fig. 6 ein Ausführungsbeispiel für den Gebrauch und den Mechanismus des Wait\_for\_Condition-Befehls, im Ablaufebenenmodell der erfindungsgemäßen industriellen Steuerung,

[0035] Fig. 7 in einer Schemadarstellung Möglichkeiten, wie der Grundtakt für die industrielle Steuerung gewonnen wird und

[0036] Fig. 8 zeigt als OO(objektorientiert)-Strukturdiagramm ein Technologiepaket, bestehend aus Code-Anteil, Parameter, Firmware-Konfiguration, Technologieobjektyp, Sprachmechanismen und Deklarationsteil.

[0037] In der Darstellung gemäß Fig. 1 sind die wesentlichen Ablaufebenen einer klassischen speicherprogrammierbaren Steuerung (SPS), angeordnet nach ihrer Priorität, gezeigt. Der Prioritätsanstieg ist dabei durch einen Pfeil symbolisiert. In der niederpriorsten Ebene werden, wie durch eine gestrichelte Linie angedeutet, zwei unterschiedliche Aufgaben, nämlich ein freier Zyklus, d. h. "Anwenderebene freier Zyklus" und eine Hintergrund-Systemebene, d. h. "Systemebene Hintergrund" abgewickelt. Der Hintergrund-Systemebene sind z. B. Kommunikationsaufgaben zugeordnet. Bei einer folgenden Anwenderebene, bezeichnet als "Anwenderebene zeitgesteuert", ist der Aufruftakt der Tasks bzw. der Programme dieser Ebene parametrierbar. Es erfolgt eine Überwachung dahingehend, ob die Bearbeitung eines Anwenderprogrammes dieser getakteten Ebene rechtzeitig abgeschlossen ist, bevor das Startereignis erneut auftritt. Läuft die Taktzeit ab, ohne dass das Anwenderprogramm der zugeordneten Ebene fertig abgearbeitet ist, wird eine entsprechende Task einer prioritätsmäßig übernächsten "Anwenderebene für asynchrone Fehler" gestartet. In dieser "Anwenderebene für asynchrone Fehler" kann der Anwender die Behandlung von Fehlerzuständen ausprogrammieren.

[0038] Auf die "Anwenderebene zeitgesteuert" folgt eine "Anwenderebene Events". Die Reaktion auf externe oder interne Ereignisse (Events) erfolgt innerhalb der "Anwender-

ebene Events". Ein typisches Beispiel für ein solches Ereignis ist das Schalten eines binären bzw. digitalen Eingangs, wodurch typischerweise ein Ereignis ausgelöst wird. In einer "Systemebene hochprior" liegen die Aufgaben des Betriebssystems, welche die Arbeitsweise der programmierbaren Steuerung (SPS) sicherstellen.

[0039] Die Darstellung gemäß Fig. 2 zeigt die wesentlichen Ablaufebenen einer Bewegungssteuerung (MC). Auch hierbei sind die einzelnen Ebenen nach ihrer Priorität hierarchisch, wie durch einen Pfeil symbolisiert, angeordnet. Eine "Systemebene Hintergrund" und eine "Anwenderebene sequentiell" haben eine gleiche Priorität, nämlich die niedrigste. Diese aufgabenmäßige Zugehörigkeit ist wie bei Fig. 1 durch eine gestrichelte Linie symbolisiert. Die Tasks der "Anwenderebene sequentiell" werden zusammen mit den Tasks der "Systemebene Hintergrund" im Round-Robin-Verfahren abgearbeitet. Typische Tasks der "Systemebene Hintergrund" sind z. B. solche für Kommunikationsaufgaben. In der "Anwenderebene sequentiell" laufen die vom Anwender programmierten Programnteile für die eigentliche Steuerungsaufgabe. Stößt die Steuerung in einem dieser Programnteile auf einen Bewegungs- oder Positionierbefehl, wird ein "Suspend" gesetzt, d. h. das Anwenderprogramm wird an dieser Stelle unterbrochen. In diesem Fall wird ein Befehl synchron genutzt. Die Abarbeitung dieses Bewegungs- oder Positionierbefehls geschieht in einer höchstpriorsten "Systemebene getaktet". Ein jeder Lageregler oder Interpolator, der in der "Systemebene getaktet" abläuft, führt diesen Bewegungs- bzw. Positionierbefehl aus. Nach Ausführung des Befehls wird in die "Anwenderebene sequentiell" zurückgesprungen und das durch "Suspend" unterbrochene Anwenderprogramm wird durch ein "Resume" an der gleichen Stelle fortgesetzt. Die "Systemebene getaktet" enthält neben den schon erwähnten Lagereglern auch den Interpolationsteil der Steuerung.

[0040] Auf die niederpriorste Ebene setzt die "Anwenderebene Events" auf. Hier sind solche Tasks untergebracht, die auf externe oder interne Ereignisse reagieren. Solche Ereignisse können beispielsweise Alarime sein.

[0041] In einer folgenden "Anwenderebene synchrongetaktet" laufen synchron getaktete Anwender-Tasks ab, z. B. Reglerfunktionalitäten. Diese Tasks sind synchronisiert zu getakteten Systemfunktionen wie zum Beispiel Interpolator, Lageregler oder zyklische Buskommunikation.

[0042] In der Darstellung gemäß Fig. 3 wird in Form eines Strukturbildes gezeigt, dass die Steuerung eines technischen Prozesses P1 über das Runtime-System RTS einer industriellen Steuerung erfolgt. Die Verbindung zwischen dem Runtime-System RTS der Steuerung und dem technischen Prozess P1 geschieht bidirektional über die Ein-/Ausgänge EA. Die Programmierung der Steuerung und damit das Festlegen des Verhaltens des Runtime-Systems RTS geschieht im Engineering-System Es. Das Engineering-System ES enthält Werkzeuge für die Konfigurierung, Projektierung und Programmierung für Maschinen bzw. für die Steuerung technischer Prozesse. Die im Engineering-System erstellten Programme werden über den Informationspfad I in das Runtime-System RTS der Steuerung übertragen. Bezüglich seiner Hardware-Ausstattung besteht ein Engineering-System ES üblicherweise aus einem Computersystem mit Graphikbildschirm (z. B. Display), Eingabehilfsmitteln (z. B. Tastatur und Maus), Prozessor, Arbeits- und Sekundärspeicher, einer Einrichtung für die Aufnahme computerlesbarer Medien (z. B. Disketten, CDs) sowie Anschlusseinheiten für einen Datenaustausch mit anderen Systemen (z. B. weiteren Computersystemen, Steuerungen für technische Prozesse) oder Medien (z. B. Internet). Eine Steuerung besteht üblicherweise aus Eingabe- und Ausgabeeinheiten, sowie aus

Prozessor und Programmspeicher. Es ist auch vorstellbar, dass die Steuerung eines technischen Prozesses P1 über mehrere Runtime-Systeme RTS von industriellen Steuerungen erfolgt.

**[0043]** Die Darstellung gemäß **Fig. 4** zeigt das Ablaufebenenmodell der industriellen Steuerung. Die Priorisierung der Ebenen wird durch einen Pfeil in Richtung zur höchsten Priorität angedeutet. Die niederpriorsten Ebenen sind die "zyklische Anwenderebene" und die "sequentielle Anwenderebene". Diese beiden Ebenen laufen mit der gleichen Priorität. Deshalb sind diese Ebenen in der Darstellung gemäß **Fig. 4** durch eine gestrichelte Linie getrennt. Die "zyklische Anwenderebene" beinhaltet die "Background Task", die zykluszeitüberwacht ist. In der "sequentuellen Anwenderebene" werden die "Motion Tasks" durchlaufen. "Motion Tasks" sind nicht zykluszeitüberwacht und dienen im Wesentlichen zur Beschreibung sequentieller Abläufe. "Motion Tasks" werden quasiparallel abgearbeitet. Generell enthalten alle Anwenderebenen eine oder mehrere Tasks. Die Tasks nehmen die Anwenderprogramme auf. Die Tasks der "zyklische Anwenderebene" und der "sequentuellen Anwenderebene" werden in einem gemeinsamen Round-Robin-Zyklus abgearbeitet.

**[0044]** Die nächstfolgende Ebene ist die "zeitgesteuerte Anwenderebene". Die Tasks dieser Ebene werden zeitgesteuert aktiviert. Die Zeitsteuerung ist einer Granularität von Millisekunden einstellbar. Auf die "zeitgesteuerte Anwenderebene" folgt die "ereignisgesteuerte Anwenderebene". In dieser Ebene werden nach Erkennen eines User Interrupts so genannte "User Interrupt Tasks" aktiviert. User Interrupt Ereignisse können als logische Verknüpfung von Prozessereignissen und/oder internen Zuständen formuliert werden.

**[0045]** Die nächsthöhere Ebene ist die "Anwenderebene für System Exceptions". In dieser "Anwenderebene für System Exceptions" werden System Interrupts überwacht, bei deren Eintreffen so genannte "Exceptions", d. h. Ausnahmefallbehandlungen, generiert werden. In der "Anwenderebene für System Exceptions" gibt es z. B. folgende Tasks, die bei Auftreten eines entsprechenden System Interrupts aktiviert werden:

- a) "Time Fault Task", die beim Ansprechen von Zeitüberwachungen aktiviert wird,
- b) "Peripheral Fault Task", die z. B. bei Prozess- und Diagnosealarmen aktiviert wird, aber auch bei Statusausfall oder Stationswiederkehr,
- c) "System Fault Task", die bei allgemeinen Systemfehlern aktiviert wird,
- d) "Program Fault Task", die bei Programmierfehlern (z. B. Division durch Null) aktiviert wird,
- e) "Time Fault Background Task", die beim Ansprechen der Zykluszeitüberwachung der Background Task aktiviert wird und
- f) "Technological Fault Task", die bei Technologiefehlern aktiviert wird.

**[0046]** Als nächstes folgt die Ebenengruppe "synchron getaktete Ebenen". Diese Ebenengruppe besitzt die höchste Priorität im Ablaufebenenmodell. Die einzelnen Ebenen dieser Ebenengruppe können untereinander weitere Priorisierungen aufweisen. Die Ebenengruppe "synchron getaktete Ebenen" besteht aus mindestens einer Systemebene und mindestens einer Anwenderebene. Die Systemebenen beinhalten die Systemfunktionen wie z. B. Lageregler oder Interpolator. In die Anwenderebenen dieser Ebenengruppe können von einem Anwender flexibel Anwenderprogramme (AP1-AP4; **Fig. 5**) zugeladen werden.

**[0047]** Für die Taktung der "synchron getakteten Ebenen"

gibt es eine Reihe unterschiedlicher Taktgenerierungsmöglichkeiten. Der Grundtakt kann z. B. aus einem internen Timer (T1; **Fig. 7**) kommen oder aus einem internen Takt (T3; **Fig. 7**) eines Kommunikationsmediums (z. B. Profibus) oder aber der Takt kann auch aus einem Prozessereignis des technologischen Prozesses abgeleitet werden. Ein solches Prozessereignis kann z. B. die Taktrate (TG; **Fig. 7**) eines Vorgangs an einer Produktionsmaschine oder Verpackungsmaschine sein. Anwenderebenen der Ebenengruppe "synchron getaktete Ebenen" können dabei basierend auf dem Grundtakt getaktet werden, sie können aber auch synchron zu einer der Systemebenen der Ebenengruppe "synchron getaktete Ebenen" laufen. Die Anwendertasks dieser zu einer Systemebene synchronen Anwenderebene haben somit eine synchrone, d. h. deterministische Beziehung zu einer vom Anwender flexibel festlegbaren Systemebene. Das hat den Vorteil, dass deterministische Reaktionen auf Systemtasks (Systemtasks laufen in den Systemebenen), die der Anwender in seinen Anwendertasks programmiert hat, die in den Anwenderebenen der Ebenengruppe "synchron getaktete Ebenen" laufen, vom System garantiert werden. Das heißt z. B., dass das System garantiert, dass diese "synchrone Anwenderebene" entsprechend beispielhaft vor dem Interpolator aktiviert wird oder aber auch vor einer beliebigen anderen Systemfunktion.

**[0048]** Die "zeitgesteuerte Anwenderebene", die "ereignisgesteuerte Anwenderebene", die "sequentielle Anwenderebene", die "zyklische Anwenderebene" sowie die "Anwenderebene für System Exceptions" sind optional.

**[0049]** Die Task der "zyklischen Anwenderebene" (Background Task) ist zykluszeitüberwacht. Die "Motion Tasks" dagegen sind nicht zykluszeitüberwacht und dienen im wesentlichen zur Beschreibung sequentieller Abläufe. Das heißt, das vorliegende Ablaufebenenmodell unterstützt einen Anwender sowohl bei der Programmierung von sequentiellen Abläufen als auch bei der Ereignisprogrammierung. Es können somit synchrone Ereignisse als auch asynchrone Ereignisse durch die Programmierung erfasst werden. In die Anwenderebenen sind die vom Anwender erstellten Anwenderprogramme (AP1-AP4; **Fig. 5**) zuladbar. Die Anwenderprogramme AP1 bis AP4 werden üblicherweise mit Hilfe einer Programmierungsumgebung des Engineering-Systems (ES **Fig. 3**) erstellt.

**[0050]** Die Darstellung gemäß **Fig. 5** zeigt ein Ausführungsbeispiel für das Zuladen von Anwenderprogrammen in die Anwenderebenen. **Fig. 5** zeigt exemplarisch eine Ausprägung von Anwenderebenen des Ablaufebenenmodells. Durch die drei Punkte am unteren Rand der Zeichnung ist dargestellt, dass auch noch weitere Anwenderebenen, aber auch Systemebenen vorhanden sein können. Die Priorisierung der Ebenen wird wie im Vorangegangenen durch einen Pfeil in Richtung zur höchsten Priorität angedeutet. Den Anwenderebenen werden die Anwenderprogramme AP1 bis AP4, am rechten Bildrand durch kleine Rechtecke angedeutet, zugeordnet. Die Zuordnung wird dargestellt durch Zuordnungspfeile ZP1 bis ZP4. In den Anwenderebenen befinden sich Tasks, die die zugeladenen Anwenderprogramme AP1 bis AP4 aufnehmen. Diese Tasks werden dann nach einer gewissen Strategie (z. B. sequentiell) durchlaufen bzw. abgearbeitet.

**[0051]** Die Darstellung gemäß **Fig. 6** zeigt ein Ausführungsbeispiel für den Gebrauch und den Mechanismus des Wait\_for\_Condition-Befehls, im Ablaufebenenmodell der erfindungsgemäßen industriellen Steuerung. Der Wait\_for\_Condition-Befehl (in **Fig. 6** dargestellt als wait\_for\_cond()) wird in dieser Darstellung exemplarisch in der "sequentuellen Anwenderebene" verwendet. Der Wait\_for\_Condition-Befehl wird in den vom Anwender er-

stellten "Motion Tasks" MT1 und MT2 verwendet, die Bestandteil der "sequentiellen Anwenderebene" sind. Die "Motion Tasks" MT1 und MT2 hängen in einem Round-Robin-Zyklus, dargestellt durch den Pfeil von MT1 nach MT2 und durch den abgewinkelten Pfeil von MT2 zu MT1. Die drei Punkte innerhalb dieses Pfeiles deuten an, dass noch weitere "Motion Tasks" im Round-Robin-Zyklus hängen können. Die "Motion Task" MT1 enthält den Wait\_for\_Condition-Befehl "wait\_for\_cond(cond\_1)", die "Motion Task" MT2 den Wait\_for\_Condition-Befehl "wait\_for\_cond(cond\_2)". Jeweils drei Punkte, die innerhalb von MT1 bzw. MT2 verwendet werden, deuten an, dass neben den beiden Wait\_for\_Condition-Befehlen und den drei Positionierbefehlen pos1() bis pos3() noch weitere Befehle in den "Motion Tasks" enthalten sein können.

[0052] Insgesamt besteht das in Fig. 6 exemplarisch dargestellte Ablaufebenenmodell eines Runtime-Systems für eine industrielle Steuerung aus folgenden Ebenen (aufgezählt von niedrigster bis höchster Priorität): "zyklische Anwenderebene", "sequentielle Anwenderebene" (die Tasks dieser beiden Ebenen haben die gleiche Priorität, dargestellt durch die gestrichelte Linie zwischen diesen Ebenen), "zeitgesteuerte Anwenderebene", "ereignisgesteuerte Anwenderebene", "Anwenderebene für Systemexceptions", "synchron getaktete Anwenderebene 2", "synchron getaktete Anwenderebene 1", "synchron getaktete Systemebene 2" und als höchstprior Ebene eine "synchron getaktete Systemebene 1".

[0053] Die Arbeitsweise des Wait\_for\_Condition-Befehls wird beispielhaft an "wait\_for\_cond(cond\_1)" aus der "Motion Task" MT1 gezeigt. Ist die "Motion Task" MT1 im Round-Robin-Zyklus an der Reihe, werden die Befehle der "Motion Task" MT1 solange bedient, bis die Zeitscheibe abgelaufen ist, oder eine Unterbrechung kommt. Trifft dies zu, wird die "Motion Task" MT2 als nächste Task im Zyklus bedient, usw. Wird in der "Motion Task" MT1 der wait\_for\_cond(cond\_1)-Befehl abgearbeitet, wird die Bedingung cond\_1 überprüft. Ist cond\_1 = true, also erfüllt, dann wird sofort der nächstfolgende Befehl pos2() ausgeführt und eventuell weitere in MT1 vorhandene Befehle sukzessive abgearbeitet, bis die Kontrolle an die nächste Task abgegeben wird.

[0054] Ist die Bedingung cond\_1 = false, also nicht erfüllt, wird sofort die "Motion Task" MT1 unterbrochen und im Round-Robin-Zyklus wird MT2 bedient. Die Bedingung cond\_1 wird aber in die "synchron getaktete Systemebene 2" eingehängt (angedeutet durch den durchgehenden abgewinkelten Pfeil vom wait\_for\_cond(cond\_1)-Befehl zu der "synchron getakteten Systemebene 2") und im Takt dieser Systemebene auf ihr Erfülltsein überprüft. Ist die Bedingung cond\_1 erfüllt, wird im Round-Robin-Zyklus die aktuelle Task verdrängt, d. h. ihr wird die Zeitscheibe entzogen und die Motion Task MT1 wird sofort unmittelbar nach dem wait\_for\_cond(cond\_1) mit dem Positionierbefehl pos2() fortgesetzt. Durch den gestrichelten Pfeil wird der Rücksprung aus der "synchron getakteten Systemebene 2" zum Positionierbefehl pos2(), d. h. zur "sequentiellen Anwenderebene" angedeutet.

[0055] Dadurch, dass bei Nichterfülltsein der Bedingung des Wait\_for\_Condition-Befehls die Überprüfung der Bedingung in einer hochprioreren "synchron getakteten Systemebene" stattfindet, und bei Erfülltsein der Bedingung sofort die unterbrochene "Motion Task" fortgesetzt wird, ist es einem Anwender möglich, bei der Programmierung von Bewegungssequenzen extrem zeitkritische Applikationen mit einfachen Sprachmitteln zu spezifizieren. Die Performance und Deterministik wird noch dadurch erhöht, dass beim Überprüfen der Bedingungen in den jeweiligen hochprioreren

"synchron getakteten Systemebenen" nur aktuell anstehende Bedingungen eingehängt und berücksichtigt werden.

[0056] Der hier beschriebene Mechanismus erfordert auch keinen expliziten Event-Handler. Der große Vorteil aus Anwendersicht liegt somit darin, dass der Anwender jetzt in einem sequentiellen Ablaufprogramm auf einer relativ niedrigen Prioritätsebene einer "Motion Task" in seinem Programmfluss hochpriorere Ereignisse mit Hilfe von Programmkonstrukten formulieren kann und nicht in ein anderes Programm wechseln muss, das er dann über andere Mechanismen (z. B. per Hand oder interruptgesteuert) auf synchrone Anwendertask abbilden muss, sondern er hat die Möglichkeit, in einem geschlossenen Anwenderprogramm diesen Zyklus "Warten auf hochprioreres Ereignis" und "hochpriorere Reaktion" auf dieses Ereignis in einem Programm geschlossen zu formulieren.

[0057] Die Bedingungen, die in einem Wait\_for\_Condition-Befehl abgefragt werden, können vom Anwender sehr flexibel und elegant formuliert werden. So kann der Anwender zur Formulierung dieser Bedingungen Programmvariablen aus einem Anwenderprogramm verwenden oder interne Größen der Steuerung oder er kann auch Prozesssignale referenzieren. Diese Größen können dann logisch, arithmetisch oder mit beliebigen Funktionen inhaltlich verknüpft werden, um daraus eine Bedingung zu formulieren. Neben dem hochprioreren Abfragen, ob die Bedingung erfüllt ist, kann man sich auch vorstellen, dass bei Erfülltsein der Bedingung auch noch dazugehöriger Programmcode, d. h. eine dahinterliegende Reaktion, die anwenderprogrammierbar ist, hochprior ausgeführt wird. Und dass erst nach der Ausführung dieses Programmcodes der Rücksprung in die niedriprior Ebene erfolgt.

[0058] Darstellung gemäß Fig. 7 zeigt in einer Schemadarstellung Möglichkeiten, wie der Grundtakt für die industrielle Steuerung gewonnen wird. Fig. 7 zeigt exemplarisch eine Kommunikationstopologie, in die die Steuerung S integriert ist. Die Steuerung S ist durch ein Rechteck dargestellt. Durch eine Anschlussleitung A2 ist die Steuerung S mit dem Bus B1 verbunden, an dem über eine Anschlussleitung A1 das externe Gerät EG hängt. Über den Bus B2 erfolgt die Verbindung zum technischen Prozess P2. Der technische Prozess P2 ist am unteren Bildrand durch ein Rechteck dargestellt. Über die Anschlussleitung A3 ist die Steuerung S mit dem Bus B2 verbunden, der wiederum über die Anschlussleitung A4 die Verbindung zum technischen Prozess P2 herstellt.

[0059] Die Generierung für den Grundtakt der Steuerung S kann aus unterschiedlichen Taktquellen erfolgen. So z. B. aus einer internen Taktquelle, dargestellt durch den internen Timer T2 der Steuerung S oder auch durch eine externe Taktquelle wie z. B. den Timer T1, der zum externen Gerät EG gehört. Als externe Taktquelle kann aber auch der Grundtakt eines Kommunikationsmediums dienen. Wenn der Bus B2 z. B. durch einen äquidistanten Profibus realisiert wird, dann kann der Takt für die Steuerung aus dem Grundtakt dieses Busses gewonnen werden. In Fig. 7 ist dies dargestellt dadurch, dass der Timer T3 direkt an der Anschlussleitung A3 positioniert ist, und diese Anschlussleitung A3 stellt die Verbindung zum Bus B2 her. Die Steuerung hängt somit als Slave am Bus und kann direkt den Bustakt verwenden. Es gibt mehrere Varianten, wie der Takt für die Steuerung aus dem Grundtakt eines Kommunikationsmediums (z. B. eines Busses) gewonnen werden kann:

Zum einen kann die Steuerung S Slave am Bus sein, die Taktinformation kommt dann von extern über den Bus. Zum anderen kann die Steuerung S Master am Bus sein. Die Taktquelle liegt in diesem Fall in der Steuerung S. Für diesen Fall existieren zwei Ausprägungen. Die Taktquelle kann in



einer Masterbusanschaltung liegen oder die Taktquelle ist in der Steuerung S. hierbei erfolgt die Einspeisung des Taktes in die Masterbusanschaltung.

[0060] Weiterhin kann als externe Taktquelle ein Taktgeber TG dienen, der im technischen Prozess P2 integriert ist. Ein Taktgeber TG in einem technischen Prozess kann z. B. der Arbeitstakt einer Produktionsmaschine oder Verpackungsmaschine sein. In der Darstellung gemäß Fig. 7 sind als Kommunikationsmedien beispielhaft Busverbindungen dargestellt. Es können aber als Kommunikationsmedien aber auch Ring-, Stern- oder andere Verbindungsarten gewählt werden, auch wireless-Verbindungen. Aus diesen Verbindungssystemen kann dann der oben genannte Grundtakt abgeleitet werden.

[0061] Die Darstellung gemäß Fig. 8 zeigt als OO-Strukturdiagramm, wobei die dazugehörigen Kardinalitäten durch eine gängige Ziffern-Notation angezeigt werden, ein Technologiepaket TP mit seinen Bestandteilen:

- a) Ablauffähige Code-Teile (Code)
- b) Parameter (PAR)
- c) Firmware-Konfiguration (FWK)
- d) Mindestens einem Technologieobjekttyp (TO)
- e) Sprachmechanismen (SPR)
- f) Deklarations- und Beschreibungsteil (ACC)

[0062] Die 1 bis n Code-Teile (z. B. C-Funktionen) werden beispielsweise für die Bewegungsführung oder die Lageregelung oder für eine andere Technologie verwendet. Die Code-Teile können unter anderem Befehle für Temperaturführung, Temperaturregelung oder für spezielle Technologien wie z. B. Pressen oder Kunststoffverarbeitung beinhalten. Wie diese Code-Teile im Ablaufebenenmodell der Steuerung in die Systemebenen eingehängt werden und in welcher Reihenfolge sie zur Abarbeitung, d. h. Ausführung, gelangen sollen, wird in der Firmware-Konfiguration FWK festgelegt. In dieser steht also die Information, in welche Systemebene ein Code-Teil integriert werden soll und wenn in einer Systemebene mehrere Code-Teile integriert sind, in welcher Reihenfolge diese Code-Teile abgearbeitet werden sollen.

[0063] Der Parameter-Teil PAR beinhaltet Oberflächen (Masken, Combo-Boxen, Regeln für die Abhängigkeit der Parameter untereinander, etc.) für das Engineering-System (ES; Fig. 3) als auch die Mechanismen für das Runtime-System (RTS; Fig. 3), die eine Parametrierung ermöglichen. Damit hat der Anwender die Möglichkeit, Instanzen von Technologieobjekttypen TO eines Technologiepaketes TP gemäß seinen Anforderungen zu parametrieren.

[0064] Mit Hilfe der 1 bis n Sprachmechanismen SPR eines Technologiepaketes TP läßt sich der Sprachvorrat des Engineering-Systems (ES; Fig. 3) um Befehle und Operatoren, die für das zugrundeliegende Technologiepaket TP mit seinen zugehörigen 1 bis n Technologieobjekttypen TO adäquat und sinnvoll sind, erweitern. Sprachmechanismen SPR müssen ins Engineering-System (ES; Fig. 3) und ins Runtime-System (RTS; Fig. 3) der Steuerung geladen werden. Nachdem solche Sprachmechanismen (z. B. "erhöhe Temperatur") im Engineering-System (ES; Fig. 3) installiert wurden, sind sie im Compiler und in der Oberfläche bzw. im Browser des Engineering-Systems (ES; Fig. 3) bekannt und können vom Anwender in seinen Anwenderprogrammen direkt verwendet werden. Durch eine Plug & Play-Technologie wird sichergestellt, daß im Engineering-System (ES; Fig. 3) bekannte Sprachmechanismen auch als ablauffähiges Codestück im Runtime-System (RTS; Fig. 3) vorhanden sind. Der Anwender verwendet also die Spezifikation der Sprachmechanismen, um die Implementierung im Runtime-

System (RTS; Fig. 3) braucht er sich nicht mehr zu kümmern. In der ACC-Komponente eines Technologiepaketes TP befindet sich die Beschreibung aller Sprachelemente, die das Technologiepaket TP enthält, die Beschreibung aller Systemvariablen und aller Typen, die im Technologiepaket TP verwendet werden. Die ACC-Komponente entspricht somit einem Deklarations- und Beschreibungsteil für das Technologiepaket TP. Diese ACC-Komponente wird in erster Linie ins Runtime-System (RTS; Fig. 3) der Steuerung geladen. Dadurch wird sichergestellt, daß sich alle Informationen bzgl. vorhandener Technologiepaketen TP und Technologieobjekttypen TO im Runtime-System der Steuerung befinden und somit der Anschluß von Bedien- und Beobachtungsgeräten (z. B. Operator Panels) sehr leicht möglich ist.

#### Patentansprüche

1. Industrielle, mit einem Runtime-System (RTS) ausgestattete Steuerung (S), insbesondere für Produktionsmaschinen, wobei das Runtime-System (RTS) ein Ablaufebenenmodell enthält, das mehrere Ablaufebenen unterschiedlichen Typs mit unterschiedlicher Priorität aufweist, wobei folgende Ablaufebenen vorgesehen sind:

- a) eine Ebenengruppe mit aus einem Grundtakt abgeleiteten synchron getakteten Ebenen, bestehend aus mindestens einer Systemebene und mindestens einer Anwenderebene, wobei die Ebenen dieser Ebenengruppe untereinander eine Priorisierung aufweisen können,
- b) einer Anwenderebene für Systemexceptions
- c) einer ereignisgesteuerten Anwenderebene,
- d) einer zeitgesteuerten Anwenderebene,
- e) einer sequentiellen Anwenderebene,
- f) einer zyklischen Anwenderebene,

wobei Anwenderebenen der Ebenengruppe a) optional synchron zu einer der Systemebenen der Ebenengruppe a) laufen können.

2. Industrielle Steuerung nach Anspruch 1, dadurch gekennzeichnet, dass der Grundtakt des Ablaufebenenmodells aus einem internen Timer (T2) oder aus einem internen Takt (T3) eines Kommunikationsmediums (B1, B2) oder aus einem externen Gerät (EG) oder von einer Größe (TG), die zum technologischen Prozess (P1, P2) gehört ableitbar ist.

3. Industrielle Steuerung nach Anspruch 1 oder Anspruch 2, dadurch gekennzeichnet, dass die ereignisgesteuerte Anwenderebene, die zeitgesteuerte Anwenderebene, die sequentielle Anwenderebene, die zyklische Anwenderebene und die Anwenderebene für Systemexceptions optional sind.

4. Industrielle Steuerung nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, dass die synchronen Ebenen zum Grundtakt übersetzt und/oder unteretzt und/oder im Verhältnis 1 : 1 getaktet sind.

5. Industrielle Steuerung nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, dass weitere priorisierende Schichtungen innerhalb der Ablaufebenen vorgesehen sind.

6. Industrielle Steuerung nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, dass optional Anwendertasks beim Systemhochlauf und/oder beim Systemrunterfahren durchlaufbar sind.

7. Industrielle Steuerung nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, dass in die Anwenderebenen Anwenderprogramme (AP1-AP4) ladbar sind.

8. Industrielle Steuerung nach einem der vorstehenden

Ansprüche, dadurch gekennzeichnet, dass in die Anwenderebenen Anwenderprogramme (AP1–AP4) ladbar sind, die je nach Typ der Anwenderebene zyklusorientiert oder sequentiell programmiert sind.

9. Industrielle Steuerung nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, dass durch das Hinzuladen von Technologiepaketen (TP) in die Systemebenen die technologische Funktionalität der Steuerung (S) erweiterbar ist.

10. Industrielle Steuerung nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, dass Mechanismen bereitgestellt sind, die es einem Anwender ermöglichen im Programmfluss auf eine beliebige Bedingung zu warten, wobei bei Erfülltsein der Bedingung der Programmfluss unmittelbar fortsetzbar ist, bei Nichterfülltsein der Bedingung der Programmfluss solange angehalten wird, bis das Erfülltsein der Bedingung festgestellt wird, wobei beim Warten auf das Erfülltsein der Bedingung die Priorität der Bedingungsüberprüfung im Vergleich zur aktuellen Taskpriorität erhöht wird.

11. Industrielle Steuerung nach Anspruch 10, dadurch gekennzeichnet, dass für die Formulierung der Bedingungen Prozesssignale und/oder interne Signale der Steuerung und/oder Variablen aus Anwenderprogrammen (AP1–AP4) verwendbar sind.

12. Industrielle Steuerung nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, dass verschiedene Ablaufebenen zur Verfügung stehen, die in Vielfachen eines Grundtaktes abgearbeitet werden, wobei technologische Funktionen den Ablaufebenen frei zugeordnet werden.

13. Industrielle Steuerung nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, dass das Runtime-System (RTS) der Steuerung (S) ein Ablaufebenenmodell enthält, das mehrere Ablaufebenen unterschiedlichen Typs mit unterschiedlicher Priorität aufweist, wobei folgende Ablaufebenen vorgesehen sind:

- a) eine Ebene, die bei Technologiefehlern aktiviert wird,
  - b) eine Ebene, die bei Zykluszeitüberwachung der Hintergrund-Tasks aktiviert wird,
  - c) eine Ebene, die bei Programmfehlern aktiviert wird,
  - d) eine Ebene, die bei Systemfehlern aktiviert wird,
  - e) eine Ebene, die bei Fehlern oder Alarmen in der Peripherie aktiviert wird,
  - f) eine Ebene, die beim Ansprechen von Zeitüberwachungen aktiviert wird,
  - g) mindestens eine Ebene, die beim Erkennen von Anwender-Interrupts aktiviert wird,
  - h) eine Interpolatorebene,
  - i) mindestens eine Ebene für Timer-Interrupts,
  - j) mindestens eine Ebene für die, vorzugsweise sequentielle, Abarbeitung von Bewegungs-Tasks und
  - k) eine Ebene für die Hintergrund-Bearbeitung,
- wobei weitere Anwendertasks beim Systemhochlauf und/oder beim Systemrunterfahren durchlaufen werden.

14. Industrielle Steuerung nach Anspruch 13, dadurch gekennzeichnet, dass der Grundtakt des Ablaufebenenmodells aus einem internen Timer (T2) oder aus einem internen Takt (T3) eines Kommunikationsmediums (B1, B2) oder aus einem externen Gerät (EG) oder von einer Größe (TG), die zum technologischen Prozess

(P1, P2) gehört abgeleitet wird.

15. Verfahren zur Erstellung eines Runtime-Systems (RTS) einer industriellen Steuerung (S) mit synchron getakteten System- und Anwenderebenen nach einem der vorstehenden Ansprüche, gekennzeichnet durch folgende aufeinander folgenden Schritte:

- a) Taktgenerierung aus einem internen HW-Takt (T2) oder einem Kommunikationssystem (B1, B2) oder einem externen Gerät (EG) oder einer Größe (TG) eines technologischen Prozesses (P1, P2),
- b) Bereitstellen von System- und Anwenderebenen für das Runtime-System (RTS) die zu diesem Grundtakt synchronisierte Ablaufebenen bereitstellen,
- c) Bereitstellen mindestens eine Anwenderebene, die synchron zu der bzw. den Systemebenen integriert ist und
- d) Programmieren der Anwenderebene bzw. der Anwenderebenen.

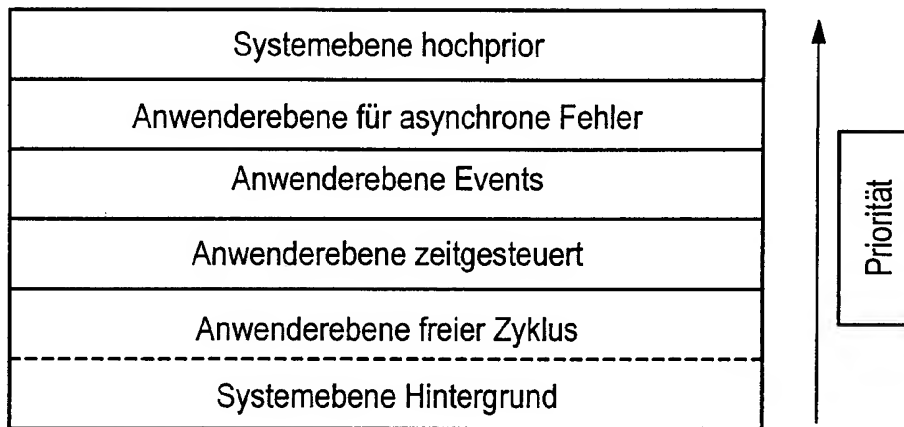
---

Hierzu 5 Seite(n) Zeichnungen

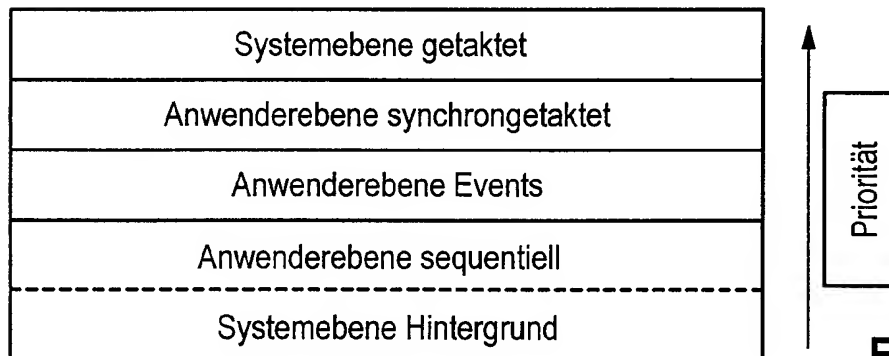
---



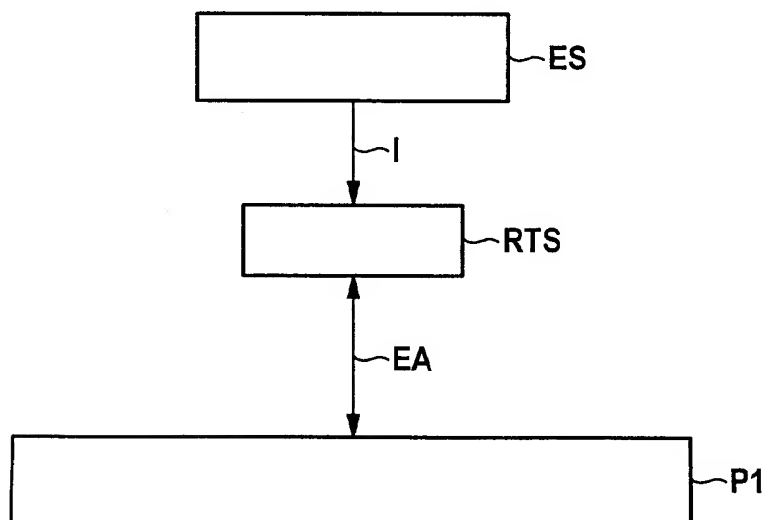
- Leerseite -



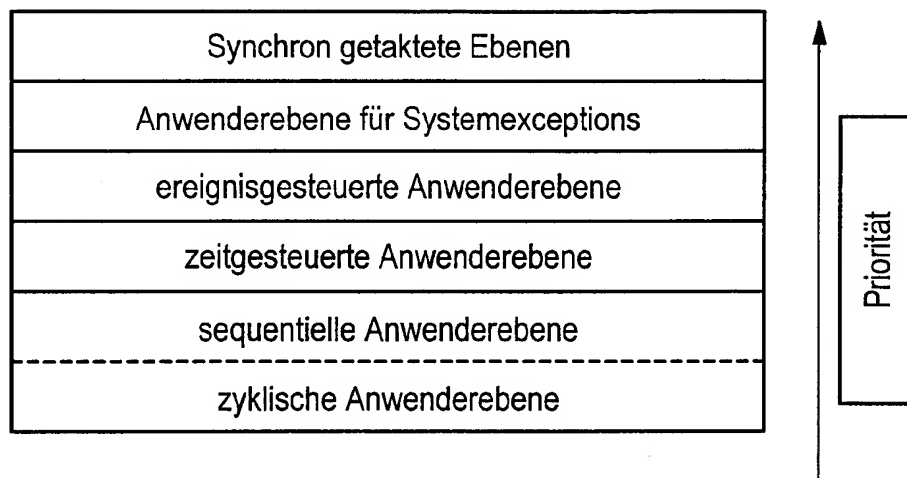
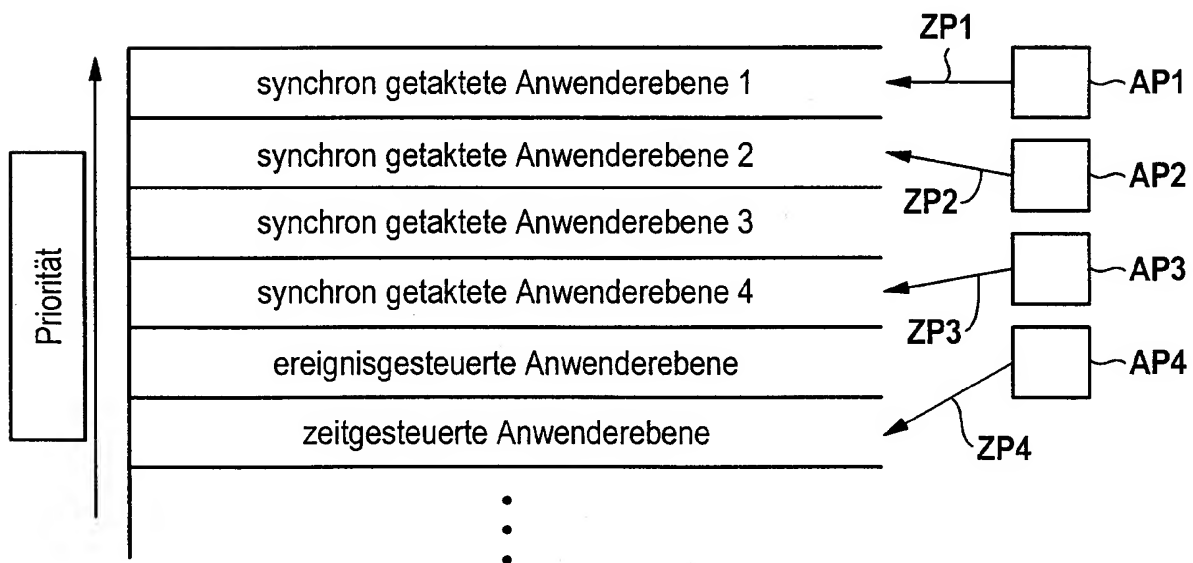
**FIG 1**

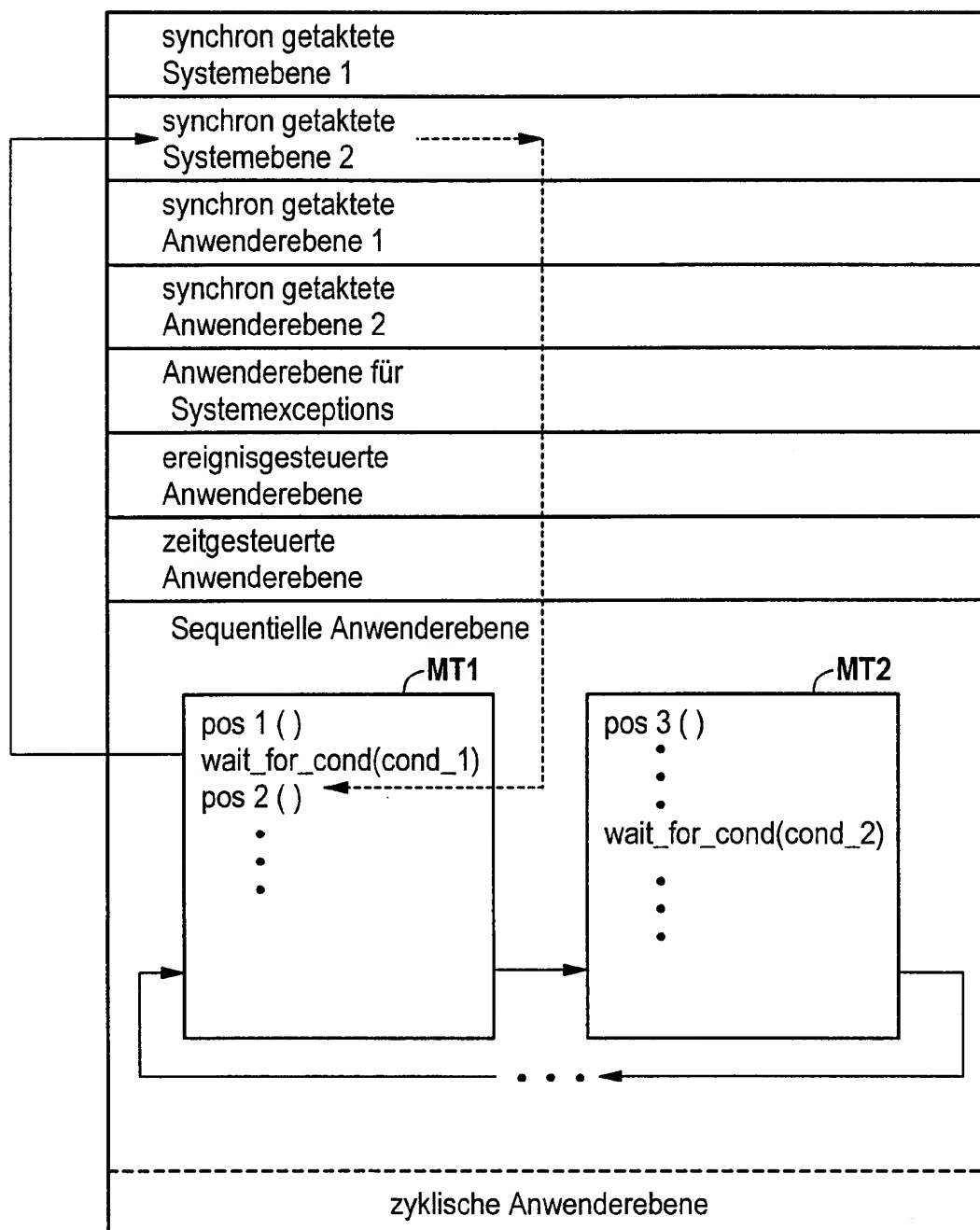


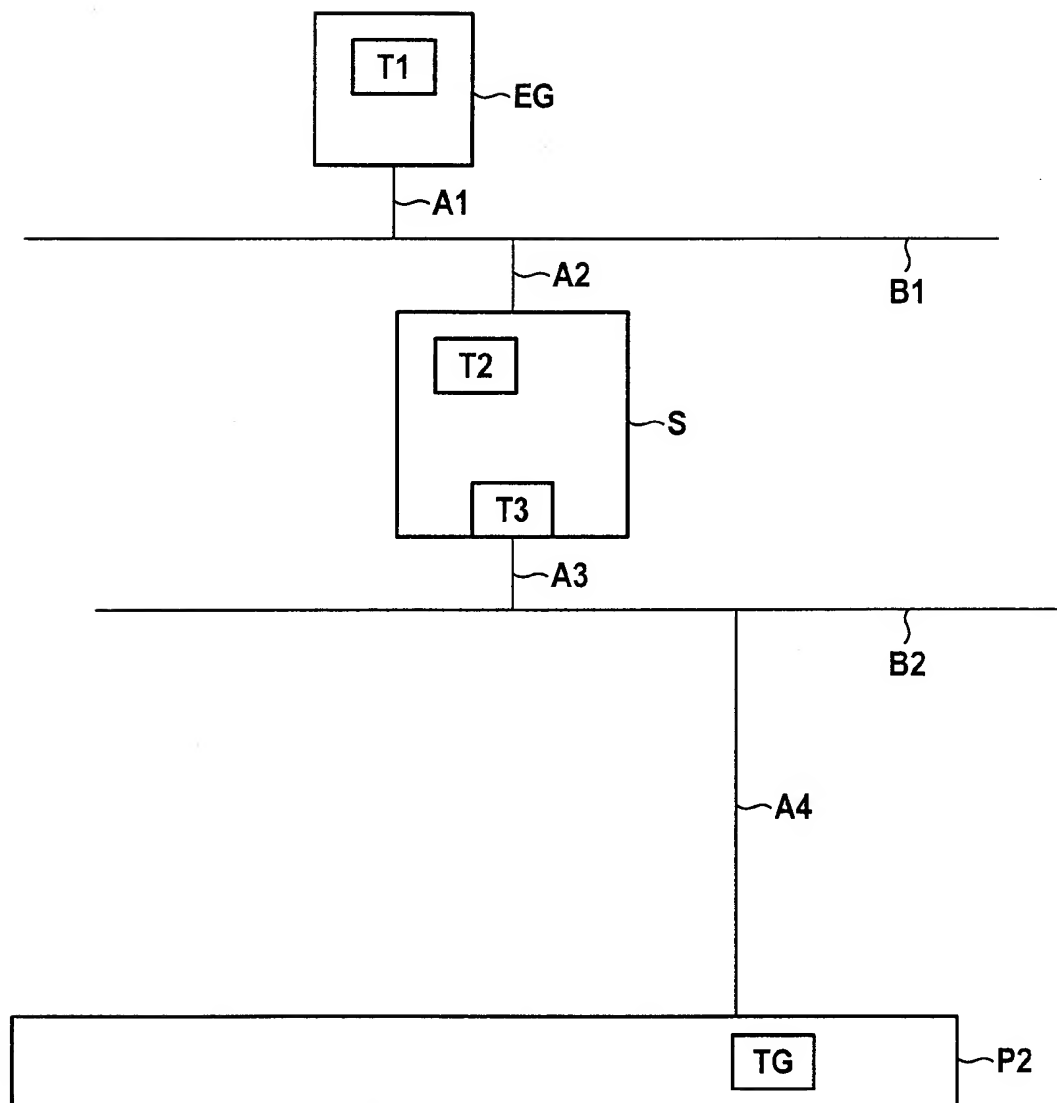
**FIG 2**



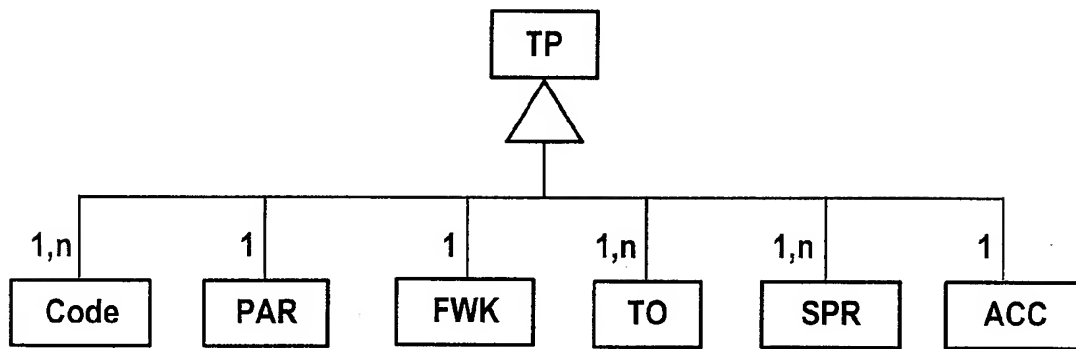
**FIG 3**

**FIG 4****FIG 5**

**FIG 6**



**FIG 7**



**FIG 8**